

BBBBBBBBBBBB		00000000		00000000		TTTTTTTTTTTT		SSSSSSSSSS
BBBBBBBBBBBB		00000000		00000000		TTTTTTTTTTTT		SSSSSSSSSS
BBBBBBBBBBBB		00000000		00000000		TTTTTTTTTTTT		SSSSSSSSSS
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBBBBBBBBBBB		000	000	000	000	TTT	SSS	
BBBBBBBBBBBB		000	000	000	000	TTT	SSS	
BBBBBBBBBBBB		000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBB	BBB	000	000	000	000	TTT	SSS	
BBBBBBBBBBBB		00000000		00000000		TTT	SSSSSSSSSS	
BBBBBBBBBBBB		00000000		00000000		TTT	SSSSSSSSSS	
BBBBBBBBBBBB		00000000		00000000		TTT	SSSSSSSSSS	

SSSSSSSS	TTTTTTTT	AAAAAA	NN	NN	DDDDDDDD	CCCCCCCC	000000	NN	NN	FFFFFFFF
SSSSSSSS	TTTTTTTT	AAAAAA	NN	NN	DDDDDDDD	CCCCCCCC	000000	NN	NN	FFFFFFFF
SS	TT	AA	NN	NN	DD	CC	00	NN	NN	FF
SS	TT	AA	NN	NN	DD	CC	00	NN	NN	FF
SS	TT	AA	NNNN	NN	DD	CC	00	NNNN	NN	FF
SSSSSS	TT	AA	NNNN	NN	DD	CC	00	NNNN	NN	FF
SSSSSS	TT	AA	NN	NN	DD	CC	00	NN	NN	FFFFFFFF
SS	TT	AA	NN	NN	DD	CC	00	NN	NN	FFFFFFFF
SS	TT	AAAAAAAA	NN	NNNN	DD	CC	00	NN	NNNN	FF
SS	TT	AAAAAAAA	NN	NNNN	DD	CC	00	NN	NNNN	FF
SS	TT	AA	NN	NN	DD	CC	00	NN	NN	FF
SSSSSSSS	TT	AA	NN	NN	DD	CC	00	NN	NN	FF
SSSSSSSS	TT	AA	NN	NN	DDDDDDDD	CCCCCCCC	000000	NN	NN	FF
SSSSSSSS	TT	AA	NN	NN	DDDDDDDD	CCCCCCCC	000000	NN	NN	FF

....
....
....
....

LL	IIIIII	SSSSSSSS
LL	IIIIII	SSSSSSSS
LL	II	SS
LL	II	SS
LL	II	SS
LL	II	SS
LL	II	SSSSSS
LL	II	SSSSSS
LL	II	SS
LL	II	SS
LL	II	SS
LL	II	SS
LLLLLLLLLL	IIIIII	SSSSSSSS
LLLLLLLLLL	IIIIII	SSSSSSSS

(2) 72
(3) 169
(5) 326

DECLARATIONS
STANDCONF - main program
Dummy entry points

```
0000 1      .TITLE  STANDCONF - MAIN PROGRAM FOR STANDALONE CONFIGURE
0000 2      .IDENT  'V04-000'
0000 3
0000 4
0000 5      *****
0000 6      *
0000 7      *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8      *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9      *  ALL RIGHTS RESERVED.
0000 10     *
0000 11     *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12     *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13     *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14     *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15     *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16     *  TRANSFERRED.
0000 17     *
0000 18     *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19     *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20     *  CORPORATION.
0000 21     *
0000 22     *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23     *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24     *
0000 25     *
0000 26     *****
0000 27
0000 28
0000 29     ++
0000 30     FACILITY:      STANDALONE CONFIGURE
0000 31
0000 32     ABSTRACT:
0000 33     THIS ROUTINE IS THE MAIN PROGRAM AND SUBROUTINES FOR STANDALONE
0000 34     CONFIGURE FOR STANDALONE VMS (I.E. STANDALONE BACKUP AND STANDALONE
0000 35     BAD).
0000 36     THIS PROCESS IS USED TO CONFIGURE ALL DISK AND PORT DRIVERS FROM
0000 37     SYSINIT. ALL MSCP- AND HSC-SERVED DEVICES ARE CONFIGURED.
0000 38
0000 39     ENVIRONMENT:  USER, EXEC, AND KERNEL MODE
0000 40
0000 41     AUTHOR:  MARYANN HINDEN,      CREATION DATE:  18-SEP-1979
0000 42            (ADAPTED FROM STASYSGEN)
0000 43
0000 44     MODIFIED BY:
0000 45
0000 46     V03-006 CWH3006      CW Hobbs      8-Oct-1983
0000 47     Properly handle the case of a null device type when displaying
0000 48     configured devices.  Call them "unknown".
0000 49
0000 50     V03-005 CWH3005      CW Hobbs      24-Sep-1983
0000 51     Add the display of configured devices so that a user has
0000 52     a better idea of what is happening.
0000 53
0000 54     V03-004 CWH3004      CW Hobbs      13-Sep-1983
0000 55     Borrow STACONFIG.MAR and produce a process suitable for
0000 56     configuring HSC and MSCP-served devices in the environment
0000 57     of standalone VMS.
```


0000	58	:	
0000	59	:	
0000	60	:	V03-002 WMC0003 Wayne Cardoza 09-Aug-1983
0000	61	:	Put D devices back.
0000	62	:	
0000	63	:	V03-001 WMC0002 Wayne Cardoza 05-Aug-1983
0000	64	:	Add lbr\$output_help.
0000	65	:	Disable autoconfigure of D devices until this module is made
0000	66	:	to deal with noncontiguous drivers.
0000	67	:	
0000	68	:	V03-001 WMC0001 Wayne Cardoza 01-Aug-1983
0000	69	:	Add BOO\$XEOPEN, BOO\$FILCLOSE, BOO\$UFOOPEN, EXE\$LOAD_CODE
0000	70	:	--

```
0000 72 .SBTTL DECLARATIONS
0000 73 :
0000 74 : INCLUDE FILES:
0000 75 :
0000 76 $ddbdef
0000 77 $dcdef
0000 78 $ipldef
0000 79 $sbdef
0000 80 $ucbdef
0000 81
0000 82 :
0000 83 : MACROS:
0000 84 :
0000 85 .macro table_entry entry_name
0000 86 table_counter = table_counter + 1
0000 87 assume dt$ 'entry_name' EQ table_counter
0000 88 .psect stacfn_tblpat exe,rd,nowrt
0000 89 td 'entry_name': .ascic /entry_name/
0000 90 .psect stacfn_tblptr exe,rd,nowrt
0000 91 .address td 'entry_name'
0000 92 .endm
0000 93
0000 94 :
0000 95 : EQUATED SYMBOLS:
0000 96 :
0000 97 :
0000 98 :
0000 99 : OWN STORAGE:
0000 100 :
0000 101
00000000 102 .PSECT BOO$SYSGEN,WRT
0000 103
00000000 0000 104 BOO$GL_CMDOPT:: .LONG 0 ; Options longword
0004 105
00000000 106 .psect stacfn_data_rw noexe,rd,wrt
0000 107
00000000 00000000 0000 108 msgdsc: .long 0,0
00000058 0008 109 msgbuf: .blkb 80
00001058 0058 110 ucb_vector: .blkl 1024 ; Remember 1000 ucb's
1058 111 ucb_end: ; Marker for end of vector table
1058 112
00000000 113 .psect stacfn_data_ro exe,rd,nowrt
0000 114
43 44 4E 41 54 53 FFFFFFFF FD050F80 0000 115 BIN_TIME: .LONG -10*1000*1000*5,-1 ; Delta for 5 seconds
00000010'010E0000' 0008 116 procnam: .ascid "STANDCONF"
46 4E 4F 0016
67 69 66 6E 6F 43 00000021'010E0000' 0019 117 ctrstr: .ascid "Configured !17<!AC$!AC!UL:!>device type !AC"
41 21 3C 37 31 21 20 20 64 65 72 75 0027
64 3E 21 3A 4C 55 21 43 41 21 24 43 0033
21 20 65 70 79 74 20 65 63 69 76 65 003F
43 41 004B
6F 63 65 72 20 74 6F 6E 20 73 69 00' 004D
64 65 7A 69 6E 67 0059
11 004D
005F
00000000 005F 119
00000000 120 table_counter = 0
121 .psect stacfn_tblptr exe,rd,nowrt
```



```
0000001A' 0000 122 disk_table:
           0000 123 .long disk_count
           0004 124 table_entry RK06
           0008 125 table_entry RK07
           000C 126 table_entry RP04
           0010 127 table_entry RP05
           0014 128 table_entry RP06
           0018 129 table_entry RM03
           001C 130 table_entry RP07
           0020 131 table_entry RP07HT
           0024 132 table_entry RL01
           0028 133 table_entry RL02
           002C 134 table_entry RX02
           0030 135 table_entry RX04
           0034 136 table_entry RM80
           0038 137 table_entry TU58
           003C 138 table_entry RM05
           0040 139 table_entry RX01
           0044 140 table_entry ML11
           0048 141 table_entry RB02
           004C 142 table_entry RB80
           0050 143 table_entry RA80
           0054 144 table_entry RA81
           0058 145 table_entry RA60
           005C 146 table_entry RC25
           0060 147 table_entry RCF25
           0064 148 table_entry RD51
           0068 149 table_entry RX50
0000001A 006C 150 disk_count = table_counter
           006C 151
00000000 006C 152 table_counter = 0
           0000006C 153 .psect stacfn_tblptr exe,rd,nowrt
           006C 154 tape_table:
00000009' 006C 155 .long tape_count
           0070 156 table_entry TE16
           0074 157 table_entry TU45
           0078 158 table_entry TU77
           007C 159 table_entry TS11
           0080 160 table_entry TU78
           0084 161 table_entry TA78
           0088 162 table_entry TU80
           008C 163 table_entry TU81
           0090 164 table_entry TA81
00000009 0094 165 tape_count = table_counter
           0094 166
           00000000 167 .psect stacfn_code exe,rd,nowrt
```

```
0000 169 .SBTTL STANDCONF - main program
0000 170 :++
0000 171 : FUNCTIONAL DESCRIPTION:
0000 172 : This is the main program for standalone CONFIGURE. It does the
0000 173 : following:
0000 174 :
0000 175 : 1) Locks the entire image into the working set.
0000 176 : 2) Call BOO$CONFIGURE, which will configure all MSCP-
0000 177 : and HSC-served devices.
0000 178 :
0000 179 : CALLING SEQUENCE:
0000 180 :
0000 181 : Called by STASYSGEN (as SYSINIT) (via the $CREPRC directive)
0000 182 :
0000 183 : INPUT PARAMETERS:
0000 184 :
0000 185 : NONE
0000 186 :
0000 187 : OUTPUT PARAMETERS:
0000 188 :
0000 189 : NONE
0000 190 :
0000 191 : ERROR INDICATIONS:
0000 192 :
0000 193 : Various errors printed on the system console
0000 194 :
0000 195 : --
000C 0000 196 :
0000 197 : .ENTRY STANDCONF,^M<R2,R3>
0002 198 :
0002 199 :
0002 200 : Load the entire image into the working set. If running from the console device, w
0002 201 : can not do any paging from the image, since the piece of media we loaded from migh
0002 202 : not be in the drive.
0002 203 :
0002 204 : $LKWSET_S INADR=BOO$GQ_LIMITS, RETADR=BOO$GQ_RETADR
0017 205 :
0017 206 :
0017 207 : Change our process name to "STANDCONF". SYSINIT (stasysgen) creates us with the
0017 208 : process name of "STANDLOAD". When SYSINIT sees our name change, it knows that we
0017 209 : have been loaded and that it is OK to request a new piece of console media.
0017 210 :
0017 211 : $SETPRN_S PRCNAM=PROCNAM
0024 212 :
0024 213 :
0024 214 : Get current values for local copy of SYSPARAM
0024 215 :
00000000*EF 00 FB 0024 216 : CALLS #0,BOO$USEACT
002B 217 :
002B 218 :
002B 219 : Set a timer for 5 seconds
002B 220 :
002B 221 : $SETIMR_S efn = #3,-
002B 222 : daytim = BIN_TIME,-
002B 223 : astadr = AST_REC
0042 224 :
0042 225 :
```


STANDCONF
V04-000

- MAIN PROGRAM FOR STANDALONE CONFIGURE 16-SEP-1984 00:03:53 VAX/VMS Macro V04-00 Page 6
STANDCONF - main program 4-SEP-1984 23:06:10 [BOOTS.SRC]STANDCONF.MAR;1 (3)

```
00000000'EF 00 FB 0042 226 ; Start threads which will configure MSCP- and HSC-served disks.
                  0042 227 ;
                  0042 228 ; CALLS #0,BOOS$CONFIGURE
                  0049 229 ;
                  0049 230 ;
                  0049 231 ; BOOS$CONFIGURE will go into a hibernate state, so we should never get here
                  0049 232 ;
04 0049 233 RET
```

```
0004 004A 235 AST_REC: .WORD ^M<R2>
      004C 236 :
      004C 237 : Go to kernel mode and print any devices configured from remote systems
      004C 238 :
      004C 239 : $cmkrnl_s routin=show_devices
      005B 240 :
      005B 241 :
      005B 242 : Reset the timer ast
      005B 243 :
      005B 244 : $SETIMR_S efn = #3,- ; Reset timer
      005B 245 : daytim = BIN_TIME,-
      005B 246 : astadr = AST_REC
      006F 247 :
04 006F 248 RET ; Dismiss AST
      0070 249 :
      0070 250 :
      0070 251 : Scan the io database and print names of devices served from remote systems
      0070 252 :
OFFC 0070 253 .entry SHOW_DEVICES,^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
      0072 254 :
      0072 255 1$: setipl #ipl$ scs ; Block SCS interrupts while scanning
5B 00000000'GF 7E 0075 256 movaq g^scs$gq_config,r11 ; Keep a pointer to the sb queue header
5A 00000000'GF 9E 007C 257 movab g^scs$ga_localsb,r10 ; Keep a pointer to the local sb
      59 6B D0 0083 258 movl (r11),r9 ; Get a pointer to the first sb in the queue
      5B 59 D1 0086 259 10$: cmpl r9,r11 ; Are we back at the sb queue header
      2E 13 0089 260 beql 60$ ; Exit when done with sb's
      5A 59 D1 008B 261 cmpl r9,r10 ; Is it our system?
      24 13 008E 262 beql 50$ ; Yes, we aren't interested
      58 44 A9 9E 0090 263 movab sb$t_nodename(r9),r8 ; Now r8 points to the nodename
      57 54 A9 D0 0094 264 movl sb$l_ddb(r9),r7 ; Get the pointer to the first ddb on the sb
      1A 13 0098 265 20$: beql 50$ ; Done with ddb's, move to next sb
      56 14 A7 9E 009A 266 movab ddb$t_name(r7),r6 ; Now r6 points to 'DBA' or something simila
      55 04 A7 D0 009E 267 movl ddb$l_ucb(r7),r5 ; Make r5 point to the first ucb on the ddb
      0B 13 00A2 268 30$: beql 40$ ; Done with ucb's, move to next ddb
      17 10 00A4 269 bsbb display_ucb ; Display one ucb
      C9 50 E8 00A6 270 blbs r0,1$ ; If we did display, start over
55 30 A5 D0 00A9 271 movl ucb$l_link(r5),r5 ; Move to the next ucb
      F3 11 00AD 272 brb 30$
      57 67 D0 00AF 273 40$: movl ddb$l_link(r7),r7 ; Move to the next ddb
      E4 11 00B2 274 brb 20$
      59 69 D0 00B4 275 50$: movl sb$l_flink(r9),r9 ; Move to the next sb
      CD 11 00B7 276 brb 10$
      04 00B9 277 60$: setipl #0 ; Restore interrupts
      00BC 278 ret
      00BD 279 :
      00BD 280 :
      00BD 281 : Display one ucb, first we check the ucb vector to see if this ucb address is
      00BD 282 : in the vector. If so, then we have already displayed this device. If not,
      00BD 283 : then put the ucb into the vector so that we won't display it again.
      00BD 284 :
      00BD 285 display_ucb:
50 00000058'EF DE 00BD 286 movl ucb_vector,r0 ; Point r0 at the vector
51 00001058'EF DE 00C4 287 movl ucb_end,r1
      51 50 D1 00CB 288 1$: cmpl r0,r1 ; At the end?
      0E 13 00CE 289 beql 3$ ; Ignore it
      55 60 D1 00D0 290 cmpl (r0),r5 ; Already displayed this one?
      09 13 00D3 291 beql 3$ ; Ignore it
```



```

50 60 D5 00D5 292      tstl      (r0)      ; Have we found an empty slot?
    08 13 00D7 293      beql      5$      ; Yes, then we must print this one
    04 C0 00D9 294      addl2     #4,r0    ; Move to the next ucb entry
    ED 11 00DC 295      brb       1$      ;
    50 D4 00DE 296 3$:  clrl      r0      ; Let them know we did not print
    05 00E0 297      rsb
    00E1 298      ;
    00E1 299      ; Haven't displayed it yet.
    00E1 300      ;
    60 55 D0 00E1 301 5$:  movl     r5,(r0)  ; Record this ucb as printed
    54 40 A5 90 00E4 302      movb     ucb$b_devclass(r5),r4 ; Grab the device class
    54 01 91 00E8 303      cmpb     #dc$_disk,r4 ; Is it a disk?
    09 12 00EB 304      bneq     10$      ; Not a disk, skip to check next
50 00000000'EF DE 00ED 305      moval     disk_table,r0 ; Make r0 point at the disk table
    0C 11 00F4 306      brb       20$      ;
    54 02 91 00F6 307 10$:  cmpb     #dc$_tape,r4 ; Is it a tape?
    E3 12 00F9 308      bneq     3$      ; We only do tapes and disks
50 0000006C'EF DE 00FB 309      moval     tape_table,r0 ; Make r0 point at the tape table
    54 41 A5 9A 0102 310 20$:  movzbl   ucb$b_devtype(r5),r4 ; Now get the device class
    0B 13 0106 311      beql      30$      ; Null type? Call it unknown.
    60 54 D1 0108 312      cmpl     r4,(r0)  ; Check against number of entries in table
    06 14 010B 313      bgtr      30$      ; Too big? Call it unknown.
    53 60 44 D0 010D 314      movl     (r0)[r4],r3 ; Get the pointer to the name
    07 11 0111 315      brb       40$      ;
53 0000004D'EF 9E 0113 316 30$:  movab    unknown,r3 ; Print the unknown device
    54 54 A5 3C 011A 317 40$:  movzwl   ucb$w_unit(r5),r4 ; Grab the unit number
00000000'EF 50 8F 9A 011E 318      movzbl   #80,msgdsc ; Set the full length of the buffer
00000004'EF 00000008'EF 9E 0126 319      movab    msgbuf,msgdsc+4 ; Move address to the descriptor
    0131 320      $fao_s   ctrstr=ctrstr,outlen=msgdsc,outbuf=msgdsc,p1=r8,p2=r6,p3=r4,p4=r3
    0152 321      setipl   #0 ; Enable interrupts while printing
    0155 322      $brdcst_s msgbuf=msgdsc ; Tell them about it
50 01 D0 0168 323      movl     #1,r0 ; Let them know we printed
    05 016B 324      rsb

```

```
016C 326 .SBTTL Dummy entry points
016C 327
016C 328 :
016C 329 : These entry points are need to resolve references to routines
016C 330 : not linked with the standalone version of CONFIG.
016C 331 :
016C 332
016C 333 EXES$LOAD CODE::
016C 334 LBR$OUTPUT_HELP::
016C 335 LIB$PUT_OUTPUT::
016C 336 LIB$GET_INPUT::
50 0000 016C 337 .WORD 0 ; ENTRY MASK
D4 016E 338 CLRL R0
04 0170 339 RET
0171 340
0171 341 BOOS$READPROMPT::
0171 342 BOOS$READFILE::
0171 343 BOOS$FILCLOSE::
0171 344 BOOS$UFOOPEN::
0171 345 BOOS$EXEOPEN::
0171 346 RIOS$OUTPUT_LINE::
50 D4 0171 347 PUTERROR::
05 0173 348 CLRL R0
0174 349 RSB
0174 350
0174 351 RIOS$GW_OUTLEN::
0174 352 RIOS$AB_BUFFER::
00000175 0174 353 .BLKB 1
0175 354
0175 355 .END STANDCONF
```


STANDCONF
Symbol table

B 8
- MAIN PROGRAM FOR STANDALONE CONFIGURE

16-SEP-1984 00:03:53 VAX/VMS Macro V04-00
4-SEP-1984 23:06:10 [BOOTS.SRC]STANDCONF.MAR;1

Page 10
(5)

```

$ST1      = 00000000
$ST2      = 00000007
AST_REC   = 0000004A R    07
BIN_TIME  = 00000000 R    04
BOOSCONFIGURE ***** X    07
BOOSEXEOPEN 00000171 RG    07
BOOSFILCLOSE 00000171 RG    07
BOOSGL_CMDOPT 00000000 RG    02
BOOSGQ_LIMITS ***** X    07
BOOSGQ_RETADR ***** X    07
BOOSREADFILE 00000171 RG    07
BOOSREADPROMPT 00000171 RG    07
BOOSUFOOPEN 00000171 RG    07
BOOSUSEACT ***** X    07
CTRSTR     00000019 R    04
DCS_DISK   = 00000001
DCS_TAPE   = 00000002
DDB$L_LINK = 00000000
DDB$L_UCB  = 00000004
DDB$T_NAME = 00000014
DISK_COUNT = 0000001A
DISK_TABLE 00000000 R    05
DISPLAY_UCB 000000BD R    07
DTS_ML1T   = 00000011
DTS_RA60   = 00000016
DTS_RA80   = 00000014
DTS_RA81   = 00000015
DTS_RB02   = 00000012
DTS_RB80   = 00000013
DTS_RC25   = 00000017
DTS_RCF25  = 00000018
DTS_RD51   = 00000019
DTS_RK06   = 00000001
DTS_RK07   = 00000002
DTS_RL01   = 00000009
DTS_RL02   = 0000000A
DTS_RM03   = 00000006
DTS_RM05   = 0000000F
DTS_RM80   = 0000000D
DTS_RP04   = 00000003
DTS_RP05   = 00000004
DTS_RP06   = 00000005
DTS_RP07   = 00000007
DTS_RP07HT = 00000008
DTS_RX01   = 00000010
DTS_RX02   = 0000000B
DTS_RX04   = 0000000C
DTS_RX50   = 0000001A
DTS_TA78   = 00000006
DTS_TA81   = 00000009
DTS_TE16   = 00000001
DTS_TS11   = 00000004
DTS_TU45   = 00000002
DTS_TU58   = 0000000E
DTS_TU77   = 00000003
DTS_TU78   = 00000005
DTS_TU80   = 00000007

```

```

DTS_TU81      = 00000008
EXESLOAD_CODE 0000016C RG    07
IPL$SCS-      = 00000008
LBR$OUTPUT_HELP 0000016C RG    07
LIB$GET_INPUT 0000016C RG    07
LIB$PUT_OUTPUT 0000016C RG    07
MSGBUF        00000008 R    03
MSGDSC        00000000 R    03
PRS_IPL       ***** X    07
PROCNAME      00000008 R    04
PUTERROR      00000171 RG    07
RISAB_BUFFER 00000174 RG    07
RISGW_OUTLEN 00000174 RG    07
RISOUTPUT_LINE 00000171 RG    07
SB$L_DDB      = 00000054
SB$L_FLINK     = 00000000
SB$T_NODENAME = 00000044
SCS$GA_LOCALSB ***** X    07
SCS$GQ_CONFIG ***** X    07
SHOW_DEVICES 00000070 RG    07
STANDCONF     00000000 RG    07
SYSSBRDCST    ***** GX    07
SYSSCMKRN     ***** GX    07
SYSSFAO       ***** X    07
SYSSLKWSET    ***** GX    07
SYSSSETIMR    ***** GX    07
SYSSSETPRN    ***** GX    07
TABLE_COUNTER = 00000009
TAPE_COUNT    = 00000009
TAPE_TABLE    0000006C R    05
TD_MC11       00000052 R    06
TD_RA60       0000006B R    06
TD_RA80       00000061 R    06
TD_RA81       00000066 R    06
TD_RB02       00000057 R    06
TD_RB80       0000005C R    06
TD_RC25       00000070 R    06
TD_RCF25      00000075 R    06
TD_RD51       0000007B R    06
TD_RK06       00000000 R    06
TD_RK07       00000005 R    06
TD_RL01       0000002A R    06
TD_RL02       0000002F R    06
TD_RM03       00000019 R    06
TD_RM05       00000048 R    06
TD_RM80       0000003E R    06
TD_RP04       0000000A R    06
TD_RP05       0000000F R    06
TD_RP06       00000014 R    06
TD_RP07       0000001E R    06
TD_RP07HT     00000023 R    06
TD_RX01       0000004D R    06
TD_RX02       00000034 R    06
TD_RX04       00000039 R    06
TD_RX50       00000080 R    06
TD_TA78       0000009E R    06
TD_TA81       000000AD R    06

```


STANDCONF
Symbol table

C 8

- MAIN PROGRAM FOR STANDALONE CONFIGURE 16-SEP-1984 00:03:53 VAX/VMS Macro V04-00 Page 11
4-SEP-1984 23:06:10 [BOOTS.SRC]STANDCONF.MAR;1 (5)

TD_TE16	00000085	R	06
TD_TS11	00000094	R	06
TD_TU45	0000008A	R	06
TD_TU58	00000043	R	06
TD_TU77	0000008F	R	06
TD_TU78	00000099	R	06
TD_TU80	000000A3	R	06
TD_TU81	000C00A8	R	06
UCB\$B_DEVCLASS	= 00000040		
UCB\$B_DEVTYPE	= 00000041		
UCB\$L_LINK	= 00000030		
UCB\$W_UNIT	= 00000054		
UCB_END	00001058	R	03
UCB_VECTOR	00000058	R	03
UNKNOWN	0000004D	R	04

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes														
. ABS .	00000000 (0.)	00 (0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE				
\$AB\$\$	00000000 (0.)	01 (1.)	NOPIC	USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE				
BOO\$SYSGEN	00000004 (4.)	02 (2.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE				
STACFN_DATA_RW	00001058 (4184.)	03 (3.)	NOPIC	USR	CON	REL	LCL	NOSHR	NOEXE	RD	WRT	NOVEC	BYTE				
STACFN_DATA_RO	0000005F (95.)	04 (4.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	NOWRT	NOVEC	BYTE				
STACFN_TBLPTR	00000094 (148.)	05 (5.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	NOWRT	NOVEC	BYTE				
STACFN_TBLDAT	000000B2 (178.)	06 (6.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	NOWRT	NOVEC	BYTE				
STACFN_CODE	00000175 (373.)	07 (7.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	NOWRT	NOVEC	BYTE				

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.09	00:00:00.45
Command processing	108	00:00:00.72	00:00:02.63
Pass 1	275	00:00:08.01	00:00:18.11
Symbol table sort	0	00:00:01.02	00:00:01.87
Pass 2	82	00:00:01.69	00:00:03.63
Symbol table output	16	00:00:00.12	00:00:00.22
Psect synopsis output	3	00:00:00.04	00:00:00.04
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	515	00:00:11.70	00:00:26.97

The working set limit was 1350 pages.
42571 bytes (84 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 684 non-local and 14 local symbols.
355 source lines were read in Pass 1, producing 32 object records in Pass 2.
26 pages of virtual memory were used to define 24 macros.

! Macro library statistics !

Macro library name	Macros defined
-----	-----
_\$255\$DUA28:[BOOTS.OBJ]BOOTS.MLB;1	0
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	5
_\$255\$DUA28:[SYSLIB]STARLET.MLB;2	15
TOTALS (all libraries)	20

786 GETS were required to define 20 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:STANDCONF/OBJ=OBJ\$:STANDCONF MSRC\$:STANDCONF/UPDATE=(ENH\$:STANDCONF)+EXECML\$/LIB+LIB\$:BOOTS.MLB/LIB

0040 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY